

What is a Lexicographical Database?

Henning Bergenholtz and Jesper Skovgård Nielsen,
Ordbogen.com. and Lemma.com., Odense, Denmark (jsn@ordbogen.com)

Abstract: Fifty years ago, no lexicographer used a database in the work process. Today, almost all dictionary projects incorporate databases. In our opinion, the optimal lexicographical database should be planned in cooperation between a lexicographer and a database specialist in each specific lexicographical project. Such cooperation will reach the highest level of success if the lexicographer has at least a basic knowledge of the topic presented in this paper: What is a database? This type of knowledge is also needed when the lexicographer describes an ongoing or a finished project. In this article, we provide the description of this type of cooperation, using the most important theoretical terms relevant in the planning of a database. It will be made clear that a lexicographical database is like any other database. The only difference is that an optimal lexicographical database is constructed to fulfil the requirements for a specific lexicographical project.

Keywords: DATABASE, RECORD, DATABASE MANAGEMENT, DESIGNING A DATABASE, USER INTERFACE, LEXICOGRAPHICAL DATABASE, DATABASE SYSTEM, DATABASE STRUCTURE, DATABASE SCHEMA

Opsomming: Wat is 'n leksikografiese databasis? Geen leksikograaf het vyftig jaar gelede 'n databasis in die werksproses gebruik nie. Vandag inkorporeer byna alle woordeboekprojekte databasisse. Na ons mening behoort die beste leksikografiese databasis beplan te word in samewerking tussen 'n leksikograaf en 'n databasisdeskundige vir elke bepaalde leksikografiese projek. So 'n samewerking sal die hoogste suksesvlak bereik as die leksikograaf ten minste 'n basiese kennis van die onderwerp van hierdie artikel het: Wat is 'n databasis? Hierdie soort kennis is ook nodig as die leksikograaf 'n voortgesette of 'n voltooide projek beskryf. In hierdie artikel verskaf ons 'n beskrywing van hierdie soort samewerking deur die belangrikste teoretiese terme te gebruik wat toepaslik is in die beplanning van 'n databasis. Dit sal duidelik gemaak word dat 'n leksikografiese databasis soortgelyk is aan enige ander databasis. Die enigste verskil is dat die beste leksikografiese databasis saamgestel is om aan die behoeftes van 'n spesifieke leksikografiese projek te voldoen.

Sleutelwoorde: DATABASIS, OPTEKENING, DATABASISBESTUUR, ONTWERP VAN 'N DATABASIS, GEBRUIKERSKOPPELVLAKE, LEKSIKOGRAFIESE DATABASIS, DATABASISSTELSEL, DATABASISSTRUKTUUR, DATABASISSKEMA

1. Databases in Lexicography

About 30 years ago, much dictionary work was done without the use of a data-

base. From the selected data, which was typically stored on written cards collected in boxes, one and only one dictionary was produced. At that time, dictionaries were mainly polyfunctional dictionaries containing almost all the selected data. Nowadays, the situation has changed. We still have many polyfunctional dictionaries and only few monofunctional dictionaries, but we will definitely produce more monofunctional information tools in the years to come. To our knowledge, there is no current dictionary project that does not use a database. This is reflected in the lexicographical literature, but in quite a disappointing way as you are never really told what a database is or how the specific database is structured. We will not go into detail with this topic and only take a few, but, as we think, representative examples.

Many contributions about lexicographical databases have in common that you can see that lexicographers use the term database without demonstrating a clear understanding of what a database really is. In reality, many, perhaps most, lexicographical discussions of databases in theoretical contributions are not really informative — sometimes they are even misleading. There are exceptions, e.g. Almind (2005), but they are few in number.

The following quotation gives the impression that the database and the dictionary is the same. This is not said explicitly, but we see no other interpretation:

The Multilingual Dictionary of Lexicographical Terms (MDLT) is an electronic dictionary available on the Internet. The content of this database and the detailed description of the entries serve many purposes. For translators, the system has term equivalents in different languages and related terms, which may help them, make adequate translations from one language into other(s). Beginners can find many interesting facts in the introductory part, which is available in both English and Russian. Transcriptions will help users to pronounce terms correctly. (Krestova and Nürnberg 2013)

If the user interface for the lexicographers is exactly the same as the user interface for the dictionary users, you could say that a database and a dictionary is the same. In reality, the user interface is not the database, but this term is often used as a practical expression for the presentation of the fields from the database. Normally, you do not have the same user interface for lexicographers and users, but you still get the impression that no real differentiation is made between a database, the dictionary planning, the dictionary production and the dictionary:

A data bank consists of information organized into records, each of which is subdivided into data fields. Creating a data bank involves a systematic process that goes from designing the form of the bank desired to implementing it on a computer. This process cannot be improvised, but must follow a number of steps. The planning starts with a definition of the major features expected of the data bank. This stage includes the following operations: a. Identification of needs, usually by means of a needs survey consisting of: identification of target users, delimitation of the needs of each user group [...] Identifying the obstacles

that might arise throughout the process: i.e. time, human resources, budget, psychological attitudes toward the new project or project change. (Cabr  1998: 169)

All of this is certainly important for the lexicographical work in a dictionary project, but not for the construction and the use of the database. The database has no direct relationship to the intended use or user group. This does not mean that the selection of different data fields is without relevance for the intended use, of course not. For example, if you do not have a field for pronunciation, you cannot produce a pronunciation dictionary or a dictionary that also informs you about the way you should pronounce a word. Of course, you have to have different fields for all the different kinds of data that you want to present in the resulting dictionary resp. dictionaries. But you can also have fields in the database that are not used at all in the lexicographical product. In principle, the database has no relationship to any intended use and user group.

The following quotation is not wrong, but misleading:

[W]e use 'database' to refer specifically to the structured collection of material assembled during the analysis process, on the basis of which final dictionary entries will be created. (Atkins and Rundell 2008: 264)

In principle, a database is an empty box created to contain data in the different fields. The database contains data, but it is not a collection of structured data. If the database description in the quotation were correct, a database could not be empty, and you would not have a database until someone puts data into the database. It is understandable that lexicographers are only interested in the data ("material"), but the data and the database should be seen as a box and the things you store in the box. The box exists both with and without content, and the same applies to a database.

2. What is a Database Really?

In this section, we will elaborate on the description of a database by providing definitions of a more technical kind, but still in a way so that non-IT specialists can understand them. We begin by defining a database and a database management system:

Database: A structured collection of values

A *structured collection* could be a table, but other options exist. For lexicographical applications, tables would be the preferred choice. A directory structure that only consists of text files could also be a database. It should be noted that the collection may be empty as a directory or a file can be empty. We shall call the structure itself the *database schema*.

By *values*, we refer to the entities that you could choose to store in the database. These might take the form of strings, numbers, dates, etc. By saving

these values in a structured manner, we allow the database management system to search for given values, retrieve all values of a certain kind or sort a given collection of values. When our values are stored in a table structure, values in the same row are related. This means that if one column consists of idioms and another consists of idiom meanings, we can search for all idioms containing the word "dog" and get the meanings for these idioms.

Database Management System (DBMS): *A software system designed to allow the definition, creation, querying, update, and administration of databases* [Wikipedia (Database), <http://en.wikipedia.org/wiki/Database>, retrieved June 2013]

To clarify this definition: If you were to choose Excel, the files created by Excel (even before the first amount of data is put in and before it is saved) would be the database, and Excel itself would be the DBMS. The structure we enforce on the data is the database schema. Granted, for most applications Excel would be a very poor DBMS, but according to our definition, it is a DBMS. In practice, however, we will choose a DBMS built only to be used as a DBMS.

We are, however, still missing a piece. The DBMS does not provide a way of accessing and modifying data for anyone but IT specialists.

User Interface (UI): *[...] is the space where interaction between humans and machines occurs* [Wikipedia (User Interface), http://en.wikipedia.org/wiki/User_Interface, retrieved June 2013]

The UI can take many forms. For example, it could be the tool that lexicographers use to feed data to the database, or it could be the representation of the data the user sees when accessing a dictionary either on a website or on her Kindle.

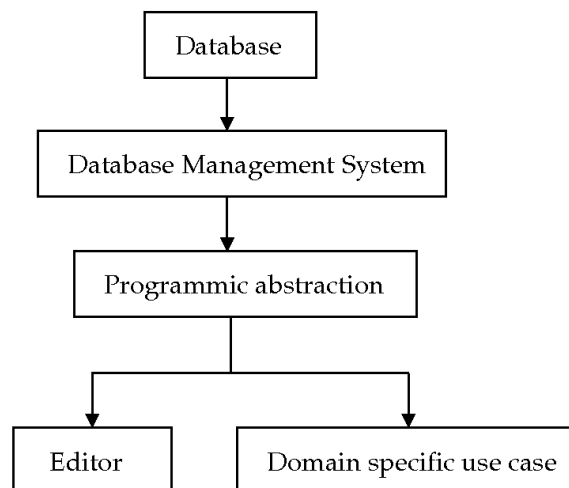
Implications

The three above-mentioned definitions together form what most people call a "database": The database provides useful data through the DBMS to the user in a UI. This more granular definition allows us to talk about the process of creating, modifying and using a database in new ways. A database contains data, and this data can be searched for and presented in different ways. For lexicographical databases, this means that a database is not a dictionary. From one database, you can create different information tools. This can be expressed more generally in the following way: On top of a database, we can build as many UIs as we like, for example one for the lexicographer to create and edit the contents of the database and one or many for the users of the database, each designed to meet a specific use. These examples are all for human utilization, but we can provide an application programming interface (API) for machine exploitation as well.

This means that a lexicographical database could have a UI for the lexicographer for updating the data. It could also have a specific UI for users trying to write a text, another one for users trying to learn more about a word, and lastly an API for a word prediction application such as the spell checker in your word processor.

However, because the database no longer "knows" what it represents, it simply corresponds to values extracted from some domain. Knowledge about what data it represents is still needed. While building the UIs, this knowledge is placed in layers of abstraction expressed in a programming language. In programming languages, we talk about objects (Lemma, Flexion, etc.). That is how we model the data in the database in useful terms.

This leaves us with a layered model of the general perception of what a database is. We call this a database system (DBS):



The most popular type of DBMS for lexicographical applications is used for relational databases. This type of database represents data in much the same way as Excel, that is, as columns of data in different tables (what Excel calls sheets). We can relate rows in different tables by defining "foreign keys", which define the relation between two tables. Some relational databases are targeted for ease of use and thus come with a pre-built UI. These include FileMaker and MS Access. These types of databases also provide a way of building customized UIs. This makes them available to non-IT specialist. As a result, they are a good match for building smaller database-driven applications where speed of development is the primary issue.

Other types of DBMSs are built to support heavy usage. These include MySQL and PostgreSQL. These DBMSs do not come with a UI, which means that more effort needs to be put into the development process. To be a feasible

option for lexicographers, a UI made in some type of programming language is needed. However, these DBMSs are much more capable of handling a large number of users. They also provide advanced ways of finding and joining data. This includes indexes and full text search. As a result, these types of DBMSs are preferable when building an online dictionary and other lexicographical applications.

Another type of database that is gaining popularity at the moment is the NoSQL database. It allows data to be saved and retrieved with lower consistency constraints. This is mainly an advantage for performance issues. This database is, however, limited in its access and storage functionality. This makes it great for storage of large amounts of data in cases where a relational database is not necessary. Examples of NoSQL databases are MongoDB, Cassandra and CouchDB.

3. The Construction and Structure of a Concrete Lexicographical Database

When we claim that a certain database can be used to produce not only one but many dictionaries, it does not mean that there are no restrictions. For instance, the result depends on the number and the content of the fields which the lexicographer has included for the lexicographical working process. For example, you cannot inform about pronunciation if you do not have a pronunciation field, and you cannot inform about the history of a word if you do not have a field for etymology. With this in mind, you can begin the construction of the database even before the lexicographer has a clear conception of the concrete dictionaries.

The fields proposed by the lexicographer do not correspond exactly to the fields made by the database specialist. The specialist may divide the fields that have been proposed by the lexicographer partly into two or more subfields.

In the following, we will describe some of the stages and decisions made in the planning of a database for a number of Spanish monolingual dictionaries. For this project, the lexicographer proposed 28 fields to be part of the database:

1. Lemma
2. Style marker to lemma
3. Sublemma
4. Homonym number
5. Polyseme number
6. Meaning
7. Lexical remark
8. Lexical remark for text production
9. Grammar, word class
10. Grammar, inflection class 1
11. Grammar, inflection class 99
12. Grammar/spelling remark

13. First reference
14. Second reference(s)
15. Collocation(s)
16. Example(s)
17. Word formation(s)
18. Synonym(s)
19. Style marker to synonym
20. Antonym(s)
21. Style marker to antonym
22. Synonym remark
23. Proverb(s)
24. Idiom(s)
25. Idiom meaning
26. Internet link
27. Dictionary grammar
28. Memo field

In practice, field number 26 may be divided in two subfields, one for the real Internet address and one for the shorter name given by the lexicographer to this address. This name will be shown in the dictionary article and used as a link to the real address.

In addition, the lexicographer proposed three kinds of buttons on the UI:

- (a) Buttons to help the lexicographers in their lexicographical work, for example a button to be used for Google searches or one to show the lexicographers the actual dictionary article in its current form.
- (b) Buttons for working with the database such as the commands: find, new lemma, delete an article, delete the data in a field, etc.
- (c) Navigation buttons for going from one of the three different pages of the UI to one of the two other pages (it is necessary to have three pages as the number of fields to be shown is too high to fit into a single screen page).

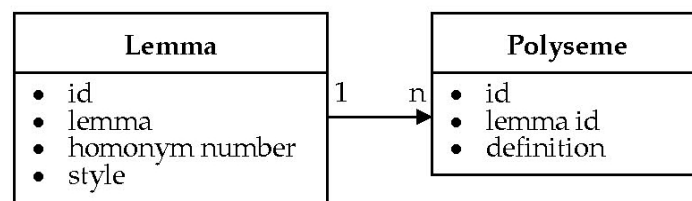
The database discussed above is made for a project with monolingual Spanish dictionaries as the outcome. These dictionaries will be somewhat similar to the six dictionaries in the ongoing Danish dictionary project that consists of one database for six dictionaries, see Bergholtz (2010, 2012, 2013).

In the following, we will describe how to transform the list of fields into a database structure. In this example, we are working on a very simple dictionary that consists of only lemma, style and meaning (polyseme). This is what the field list looks like:

- Lemma
- Homonym number
- Style

- Definition
- Polyseme number

The first thing to do is to group relevant fields together as they need to go into the same table. We might argue that everything should be put into the same table, but this leaves us with the question of how to handle multiple polysemes. Instead, we place the polysemes into their own table. We give each lemma a unique id. For every polyseme, we add the lemma id which links it to the lemma to which it belongs. We call the lemma id in the polyseme table a *foreign key*. The database schema looks like this:



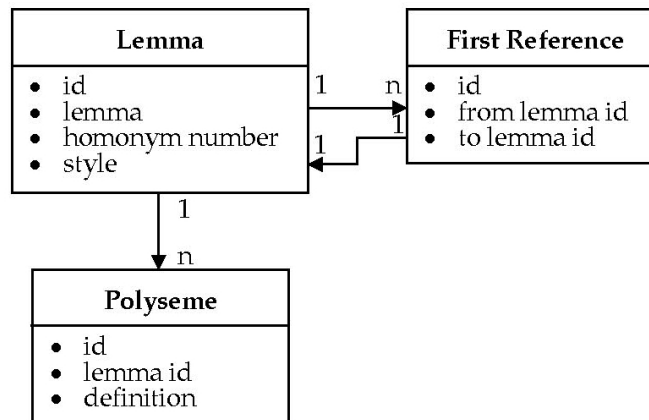
By having a foreign key in the polyseme table, we relate the lemma and the polyseme, in effect creating a one-to-many relationship. This is a technique used in many places in a dictionary. Collocations, synonyms and the like are linked to polysemes in the same manner, that is, by adding polyseme ids to collocations and synonyms. Notice how our use of the polyseme table differs from how we normally talk about polysemes. Normally, a lemma would have zero, two or more polysemes, never one. However, it makes sense to use the polyseme table to store the definition for a record, even if we consider it a lemma record with only one meaning. It should also be noticed how the fields in the database diverge from the fields provided by the lexicographer.

Using Relation Tables

We want to make "first references", in effect linking two lemmata to each other by referring the user from one lemma to another. Now, our field list looks like this:

- Lemma
- Homonym number
- Style
- First reference
- Definition
- Polyseme number

We do this by making a link table:



The field "from lemma id" is the id of the lemma we want to link from. The field "to lemma id" is the id of the lemma we want to link to. In this way, a lemma can link to one or more lemmata. This allows us to make stable connections between lemmata. Had we simply used the lemma and its homonym number to make the link, the connection would not be stable. What if the spelling changed? Or what if the homonym number changed? With this structure, we do not need to worry about that.

We can use the same technique for second references, that is, by linking polysemes. Again, we see the field lists diverge even more. We also see that the way we represent data in the database is widely different from the way it is presented to users of the database via the UI. We cannot show simple ids of the lemmata when we present links between lemmata. For the users of the dictionary, we present links to the articles. For the lexicographers, we present lists of lemmas, which enable them to link lemmata.

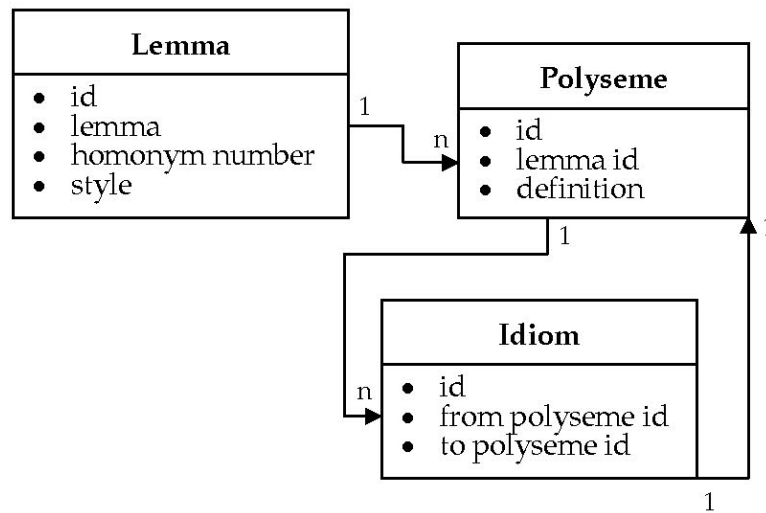
A More Complex Example

If we expand our field list with idioms, we get the following list:

- Lemma
- Homonym number
- Style
- Definition
- Polyseme number
- First reference
- Idiom
- Idiom meaning
- Style

We already know that we want to be able to represent idioms as individual

articles at a later stage. We notice how an idiom can be represented exactly like a lemma with a polyseme. Given that we want to transform idioms into individual articles at a later stage, we can use the lemma and polyseme table to store idioms in the following way:



Firstly, notice that we have made an extra field in the lemma table for the new lemma types called idiom lemma resp. proverb lemma. Because we are storing both lemmata and idioms, we need to know which is which. Secondly, we have made a new table that links one polyseme to another. The "from polyseme id" is the polyseme id which has the idiom. The "to polyseme id" is the polyseme id of the polyseme with the definition of the idiom. Using the lemma id in the polyseme table, we can get the lemma field which is our idiom. This "abuse" of our database solves our problem of letting idioms become individual articles at a later stage while still making us able to link them to polysemes as regular idioms.

This structure also allows us gradually to let the idioms grow into articles and make links directly to them simply by changing their type from idiom to lemma when they become articles themselves. Again, we see how our database has diverged even further from the field list. This is the result of the functional demands we set up for the database. These are hard to express in a field list, and the database expert on the team needs to be aware of these things by asking the right questions to the lexicographer. Furthermore, the UI would also need to display data in a very different manner compared to how it is represented in the database. The abstraction in the database should be hidden from all users of the database.

4. Lexicographical Database

Every database is constructed for use within a narrow or a broad field. If a database is used for its designed function, we can speak about a genuine use of the database; if not, we speak about a non-genuine use. For example, Excel is conceived as a spread sheet processor; this is the genuine use of Excel. But Excel can also be used and is used in dictionary projects; this is a non-genuine use of Excel. Different databases designed to be used in lexicographical work will be different according to the need of the lexicographical project. From these considerations, we can conclude:

Database: A structured collection of values

Lexicographical Database (LDB): A database constructed to contain lexicographical data

References

- Almind, R. 2005. Designing Internet Dictionaries. *Hermes, Journal of Linguistics* 34: 37-54.
- Atkins, B.T.S. and M. Rundell. 2008. *The Oxford Guide to Practical Lexicography*. Oxford/New York: Oxford University Press.
- Bergenholtz, H. 2010. Needs-Adapted Data Access and Data Presentation: Bergenholtz, H. 2010. *Doctorado Honoris Causa del Excmo. Sr. D. Henning Bergenholtz*: 41-57. Valladolid.
- Bergenholtz, H. 2012. Conceptions for Different Types of Language Tools. Ndinga-Koumba-Binza, Hugues Steve and Sonja E. Bosch. (Eds.). 2012. *Language Science and Language Technology in Africa: Festschrift for Justus C. Roux*: 328-339. Stellenbosch: SUN PReSS.
- Bergenholtz, H. 2013. Wortbildungsangaben als Hilfe für den Zugriff auf andere Datentypen und als Hilfe bei kommunikativen und kognitiven Informationsbedürfnissen. Klosa, Annette (Ed.). 2013. *Wortbildung im elektronischen Wörterbuch*: 133-156. Tübingen: Gunther Narr.
- Cabré, M.T. 1998. *Terminology. Theory, Methods and Applications*. Amsterdam/Philadelphia: John Benjamins.
- Krestova, S. and P.J. Nürnberg. 2013. *Multilingual Dictionary of Lexicographical Terms*. (accessed 23 May 2013 http://www.papillon-dictionary.org/static/info_media/42808981.pdf.)